IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:	Atty. Docket No.: 007287.00018
Dan Kikinis	
Serial No.: 10/071,091	Group Art Unit: 2424
Filed: February 7, 2002	Examiner: Shang, Annan Q.
For: A METHOD AND SYSTEM FOR IMPLEMENTING AN ELECTRONIC PROGRAMMING GUIDE	Confirmation No.: 1032

APPEAL BRIEF

U.S. Patent and Trademark Office
Customer Service Window
Mail Stop - Appeal
Randolph Building
401 Dulany Street
Alexandria, VA 22314

Sir:

This is an Appeal Brief filed in support of Appellant's Notice of Appeal filed May 5, 2009. Appeal is taken from the Final Office Action mailed February 5, 2009 (hereinafter "Final Office Action") and the Notice of Panel Decision from Pre-Appeal Brief Review mailed July 20, 2009.

General Authorization of Payment of Fees

Please charge any fees to Deposit Account No. 19-0733. In addition, any necessary extensions of time are hereby requested.

TABLE OF CONTENTS

REAL PARTY IN INTEREST
37 C.F.R. § 41.37(c)(1)(i)

The owner of this application, and the real party in interest, is JLB Ventures LLC.

RELATED APPEALS AND INTERFERENCES
37 C.F.R. § 41.37(c)(1)(ii)

There are no related appeals or interferences.

STATUS OF CLAIMS
37 C.F.R. § 41.37(c)(1)(iii)

Claims 1-14, 29-42 and 67-70 are rejected and presently appealed.

Claims 57-66 have been withdrawn from consideration.

Claims 15-28 and 43-56 have been canceled.

STATUS OF AMENDMENTS
37 C.F.R. § 41.37(c)(1)(iv)

No amendments have been made subsequent to final rejection.

SUMMARY OF CLAIMED SUBJECT MATTER
37 C.F.R. § 41.37(c)(1)(v)

In making reference herein to various embodiments in the specification text and/or drawings to explain the claimed invention, Appellant does not intend to limit the claims to those embodiments; all references to the filed specification and drawings are illustrative unless otherwise explicitly stated. Appellant refers to the patent application publication, U.S. Patent Pub. No. 2002/0129370, for the cited support.

Independent claim 1 is directed to a method for implementing an electronic programming guide (EPG). *U.S. Patent Publication No. 2002/0129370*, Abstract, ¶¶ [0002], [0017], [0018]. The method includes receiving programming information from a source. *Id.*, ¶¶ [0017], [0018], and [0024]; Fig. 2. The method further includes storing the received programming information, in its entirety, in a data storage area. *Id.*, ¶ [0017], [0024]; Figs. 2 and 6. The method further includes responsive to the received programming information being stored in its entirety, continuously scanning the data storage area, in a cyclical manner, to identify and remove data entries meeting a first predetermined criterion. *Id.*, ¶¶ [0017], [0018], [0020] and [0024]; Figs. 2

and 6. The method further includes, responsive to storing the received programming information in its entirety, partitioning the data storage area into a plurality of discrete storage areas. *Id.*, ¶ [0020]; Fig. 4. The method further includes distributing stored programming information to each discrete storage area based on a second predefined criterion. *Id.*, ¶¶ [0020], [0021]; Figs. 4 and 5.

Independent claim 29 is directed to one or more computer-readable media storing instructions that, when executed by a processor, cause the processor to perform a method. *Id.*, ¶ [0025]. The method includes receiving programming information from a source. *Id.*, ¶¶ [0017], [0018], and [0024]; Fig. 2. The method further includes storing the received programming information, in its entirety, in a data storage area. *Id.*, ¶¶ [0017], [0024]; Figs. 2 and 6. The method further includes, responsive to the received programming information being stored in its entirety, continuously scanning the data storage area, in a cyclical manner, to identify and remove data entries meeting a first predetermined criterion. *Id.*, ¶¶ [0017], [0018], [0020], and [0024]; Figs. 2 and 6. The method further includes, responsive to storing the received programming information in its entirety, partitioning the data storage area into a plurality of discrete storage areas. *Id.*, ¶ [0020]; Fig. 4. The method further includes distributing stored programming information to each discrete storage area based on a second predefined criterion. *Id.*, ¶¶ [0020], [0021]; Figs. 4 and 5.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL
37 C.F.R. § 41.37(c)(1)(vi)

- Claims 1-8, 12-14, 29-36, 40-42, 67 and 68 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,728,966 to Arsenault *et al.* ("Arsenault").

- Claims 9-11, 37-39, 69 and 70 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Arsenault in view of U.S. Patent No. 5,155,831 to Emma *et al.* ("Emma").

ARGUMENT
37 C.F.R. § 41.37(c)(1)(vii)

A.   Rejection of Claims 1-8, 12-14, 29-36, 40-42, 67 and 68 Over Arsenault

1. Independent Claim 1

Independent claim 1recites, among other features, "a method for implementing an electronic program guide, the method comprising receiving programming information from a source, storing the received programming information, *in its entirety*, in a data storage area, responsive to the received programming information being stored in its entirety, continuously scanning the data storage area, in a cyclical manner, to identify and remove data entries meeting a first predetermined criterion..." (Emphasis added). Arsenault fails to teach or suggest the features of claim 1.

Arsenault describes an electronic programming guide with a plurality of linked lists that are used to remove redundancy, increase search speed and allow dynamic categories. Abstract. The system of Arsenault receives a stream of digitized data packets containing information and the packets are *filtered out* to remove any packets that are not currently of interest. See col. 6, lines 5-11 (emphasis added). That is, the data is filtered *prior to* being stored.

In the Final Office Action, the Office asserts, at p. 2, that various portions of Arsenault teach or suggest the data being stored in its entirety. Appellant respectfully disagrees. At most, the cited portions of Arsenault teach or suggest content packets, created from *filtered* information, that are *stored as they are created. See* col. 6, line 55 – col. 7, line 46. Moreover, the packets that are of interest *may be intermediately stored* in system RAM. Col. 6, lines 11-28. There is absolutely no teaching or suggestion of data being stored in its entirety, let alone a method wherein, responsive to the data being stored *in its entirety*, the data storage area is continuously scanned, in a cyclical manner, to identify and remove data entries meeting a first predetermined criterion. Rather, Arsenault describes a system and method wherein incoming data is sorted upon entry to determine what will be saved and what will be discarded.

Further, because Arsenault fails to teach or suggest storing data in its entirety, it follows that Arsenault also fails to teach or suggest, responsive to storing the received programming information in its entirety, partitioning the data storage area into a plurality of discrete storage areas, and distributing stored programming information to each discrete storage area based on a predefined criterion, as recited in claim 1. Assuming, without conceding, that Arsenault describes discrete storage areas, Arsenault clearly fails to teach or suggest partitioning the storage area *responsive to* storing the programming information *in its entirety*. In fact, Arsenault

teaches away from such an arrangement because the incoming data of Arsenault is sorted and filtered prior to being saved.

For at least the above-discussed reasons, claim 1 is allowable over the cited reference.

2.   *Independent Claim 29*

Independent claim 29 recites features similar to claim 1 and is allowable for at least the same reasons as discussed above with respect to claim 1, and further in view of the additional novel and non-obvious features recited therein.

3. *Dependent Claims 2, 4-8, and 12-14*

Claims 2, 4-8, and 12-14 depend from claim 1and are allowable over the cited reference for at least the same reasons as discussed above with respect to claim 1.

4. *Dependent Claims 30, 32-36, and 40-42*

Claims 30, 32-36, and 40-42 depend from claim 29 and are allowable for at least the same reasons as discussed above with respect to claim 29.

5. *Dependent Claims 3 and 31*

Claims 3 and 31 depend from claims 1 and 29, respectively, and are thus allowable for at least the same reasons as their respective base claims.

Further, claims 3 and 31 recite, among other features, programming information further comprising tokens, including *compressed forms* of the information about the individual programs, used to describe individual programs and a meaning associated with the tokens. Arsenault fails to teach or suggest this feature. Instead, Arsenault describes content records that are accompanied by a title and a label. Col. 6, lines 55-63. Arsenault fails to teach or suggest tokens that include *compressed forms* of information about individual programs. Further, there is no teaching or suggestion of a *meaning* associated with the tokens. Rather, the records of Arsenault merely include a title and a label. Accordingly, Appellant respectfully submits that claims 3 and 31 are allowable over the cited reference.

6. Claims 67-68

Claims 67-68 depend from claim 1and are thus allowable for at least the same reasons as discussed above with respect to claim 1.

Further, claims 67 and 68 recite wherein the meaning associated with the tokens is stored in a token dictionary and the token dictionary is modifiable, respectively. There is absolutely no teaching or suggestion in Arsenault of a token having any meaning, let alone a meaning stored in a token dictionary. Rather, the cited portions of Arsenault merely describe titles anlabels. There is no teaching or suggestion of a meaning associated with the title or label, let alone a meaning stored in a modifiable dictionary. Accordingly, Applicants respectfully assert that claims 67 and 68 are patentably distinct from Arsenault.

B.   Rejection of Claims 9-11, 37-39, 69, 70 over Arsenault in view Emma

Claims 9-11 and 69 depend from claim 1, and claims 37-39 and 70 depend from claim 29; therefore these claims are allowable for at least the same reasons as their respective base claims. The addition of Emma fails to cure the deficiencies of Arsenault with respect to claims 1 and 29. For instance, Emma fails to teach or suggest a method wherein received programming information is stored *in its entirety* and, responsive to storing the received programming information in its entirety, continuously scanning the data storage area to identify and remove data entries meeting a first predetermined criterion. Accordingly, because neither Arsenault nor Emma, alone or in combination, teaches or suggests the features of independent claims 1 and 29, Appellant respectfully submits that claims 9-11, 37-39, 69 and 70 are allowable over the cited combination of references.

CONCLUSION

For all of the foregoing reasons, Appellant respectfully submits that the final rejection of claims 1-14, 29-42, and 67-70 is improper and should be reversed.

Respectfully submitted,

BANNER & WITCOFF, LTD.

Dated: August 20, 2009   By: /Elizabeth A. Almeter/

Elizabeth A. Almeter
Reg. No. 57,019
1100 13th Street, N.W.
Washington, D.C. 20005
Tel: (202) 824-3000
Fax: (202) 824-3001

Claims involved in the appeal:

Claim 1: A method for implementing an electronic program guide, the method comprising:

receiving programming information from a source;

storing the received programming information, in its entirety, in a data storage area;

responsive to the received programming information being stored in its entirety, continuously scanning the data storage area, in a cyclical manner, to identify and remove data entries meeting a first predetermined criterion;

responsive to storing the received programming information in its entirety, partitioning the data storage area into a plurality of discrete storage areas; and

distributing stored programming information to each discrete storage area based on a second predefined criterion.

Claim 2: The method of claim 1, wherein the programming information comprises information about individual programs.

Claim 3: The method of claim 2, wherein the programming information further comprises tokens, including compressed forms of the information about individual programs, used to describe the individual programs and a meaning associated with the tokens.

Claim 4: The method of claim 2, wherein the predefined criterion comprises a temporal relationship between the individual programs in the received program information.

Claim 5: The method of claim 3, wherein the predefined criterion comprises a numeric relationship between token numbers associated with the tokens.

Claim 6: The method of claim 1, wherein a size of each data storage area is selected to store program information about programs to be broadcast over a defined time interval.

Claim 7: The method of claim 1, further comprising referencing the information stored in each discrete storage area using a storage area identifier to identify the information within a storage area and an index of storage area identifiers.

Claim 8: The method of claim 7, wherein the storage area identifiers form a pointer chain.

Claim 9: The method of claim 7, wherein each discrete storage area which stores programming information no longer required is referenced by an empty identifier indicating that said discrete storage area is available for storing new information.

Claim 10: The method of claim 9, further comprising periodically determining if the programming information stored in each discrete storage area is relevant; and marking those storage areas containing programming information that is no longer relevant with the empty identifier.

Claim 11: The method of claim 10, wherein determining if the programming information stored in each discrete storage area is relevant comprises checking if the programming information is current or not.

Claim 12: The method of claim 1, further comprising determining that specific programming information is required; and requesting said specific programming information from the source.

Claim 13: The method of claim 12, wherein determining that specific programming information is required comprises checking if a user has input a request for specific programming information.

Claim 14: The method of claim 12, wherein determining that specific programming information is required comprises checking whether the programming information stored in the discrete storage areas is incomplete for want of specific programming information.

Claim 29: One or more computer-readable media storing instructions that, when executed by a processor, cause the processor to perform a method, comprising:

receiving programming information from a source;

storing the received programming information, in its entirety, in a data storage area;

responsive to the received programming information being stored in its entirety, continuously scanning the data storage area, in a cyclical manner, to identify and remove data entries meeting a first predetermined criterion;

responsive to storing the received programming information in its entirety, partitioning the data storage area into a plurality of discrete storage areas; and

distributing stored programming information to each discrete storage area based on a second predefined criterion.

Claim 30: The one or more computer readable media of claim 29, wherein the programming information comprises information about individual programs.

Claim 31: The one or more computer readable media of claim 30, wherein the programming information further comprises tokens, including compressed forms of the information about individual programs, used to describe the individual program and a meaning associated with the tokens.

Claim 32: The one or more computer readable media of claim 30, wherein the predefined criterion comprises a temporal relationship between the individual programs in the received program information.

Claim 33: The one or more computer readable media of claim 31, wherein the predefined criterion comprises a numeric relationship between token numbers associated with the tokens.

Claim 34: The one or more computer readable media of claim 29, wherein a size of each data storage area is selected to store program information about programs to be broadcast over a defined time interval.

Claim 35: The one or more computer readable media of claim 29, wherein the method further comprises referencing the information stored in each discrete storage area using a storage area identifier to identify the information within a storage area and an index of storage area identifiers.

Claim 36: The one or more computer readable media of claim 35, wherein the storage area identifiers form a pointer chain.

Claim 37: The one or more computer readable media of claim 35, wherein each discrete storage area which stores programming information no longer required is referenced by an empty identifier indicating that said discrete storage area is available for storing new information.

Claim 38: The one or more computer readable media of claim 37, wherein the method further comprises periodically determining if the programming information stored in each discrete storage area is relevant; and marking those storage areas containing programming information that is no longer relevant with the empty identifier.

Claim 39: The one or more computer readable media of claim 38, determining if the programming information stored in each discrete storage identifier is relevant comprises checking if the programming information is current or not.

Claim 40: The one or more computer readable media of claim 29, wherein the method further comprises determining that specific programming information is required and requesting said specific programming information from the source.

Claim 41: The one or more computer readable media of claim 40, wherein determining the specific programming information is required comprises checking if a user has input a request for specific programming information.

Claim 42: The one or more computer readable media of claim 40, wherein determining that specific programming information is required comprises checking whether the programming

information stored in the discrete storage area is incomplete for want of specific programming information.

Claim 57: (Withdrawn) A method for implementing an electronic program guide, the method comprising:

receiving programming information from a source;

storing the received programming information, in its entirety, in a data storage area;

cyclically scanning the stored programming information to identify missing programming information based at least one predefined criterion;

transmitting a request to the source for the missing programming information, the request including information identifying the missing programming information; and

receiving the missing programming information from the source.

Claim 58: (Withdrawn) The method of claim 57, wherein the step of receiving programming information from the source further includes receiving tokens used to describe individual programs and a meaning associated with the tokens.

Claim 59: (Withdrawn) The method of claim 58, wherein the step of receiving tokens further includes receiving a token dictionary in which the tokens and the associated meaning are stored.

Claim 60: (Withdrawn) The method of claim 59, wherein the token dictionary can be modified.

Claim 61: (Withdrawn) The method of claim 57, wherein the at least one predefined criterion includes a temporal relationship between individual programs in the received programming information.

Claim 62: (Withdrawn) One or more computer-readable media storing instructions that, when executed by a processor cause the processor to perform a method, comprising:

receiving programming information from a source;

storing the received programming information, in its entirety, in a data storage area;

cyclically scanning the stored programming information to identify missing programming information based at least one predefined criterion;

transmitting a request to the source for the missing programming information, the request including information identifying the missing programming information; and

receiving the missing programming information from the source.

Claim 63: (Withdrawn) The one or more computer readable media of claim 62, wherein the step of receiving programming information from the source further includes receiving tokens used to describe individual programs and a meaning associated with the tokens.

Claim 64: (Withdrawn) The one or more computer readable media of claim 63, wherein the step of receiving tokens further includes receiving a token dictionary in which the tokens and the associated meaning are stored.

Claim 65: (Withdrawn) The one or more computer readable media of claim 64, wherein the token dictionary can be modified.

Claim 66: (Withdrawn) The one or more computer readable media of claim 62, wherein the at least one predefined criterion includes a temporal relationship between individual programs in the received programming information.

Claim 67: The method of claim 3, wherein the meaning associated with the tokens is stored in a token dictionary.

Claim 68: The method of claim 67, wherein the token dictionary is modifiable.

Claim 69: The method of claim 1, wherein the first predetermined criterion is the data is expired.

Claim 70: The one or more computer readable media of claim 29, wherein the first predetermined criterion is the data is expired.

<u>EVIDENCE APPENDIX</u>

37 C.F.R. § 41.37(c)(1)(ix)

NONE.

<u>RELATED PROCEEDINGS APPENDIX</u>
37 C.F.R. § 41.37(c)(1)(x)

NONE.